



EXPERIMENT - 21

Object Oriented Programming Lab

Aim

Design three classes STUDENT, EXAM, and RESULT. The STUDENT class has data members such as roll no, name. Create a class EXAM by inheriting the STUDENT class. The EXAM class adds data members representing the marks scored in six subjects. Derive the RESULT from the EXAM class and has its own data members such as total marks. WAP to model this relationship.

Syeda Reeha Quasar

14114802719

4C7

EXPERIMENT – 21

Aim:

Design three classes STUDENT, EXAM, and RESULT. The STUDENT class has data members such as roll no, name. Create a class EXAM by inheriting the STUDENT class. The EXAM class adds data members representing the marks scored in six subjects. Derive the RESULT from the EXAM class and has its own data members such as total marks. WAP to model this relationship.

Source Code:

```
#include <iostream>
#include <string>
using namespace std;

class student
{
    string name;
    int rollNo;

public:
    void input()
    {
        cout << "\nEnter name: ";
        cin >> name;
        cout << "\nEnter roll no: ";
        cin >> rollNo;
    }
    void output()
    {
        cout << "\nNAME      : " << name;
        cout << "\nROLL NO. : " << rollNo;
    }
};

class exam : public student
{
    float marks[6];

public:
```

```

        void enterData();
        float add();
};

void exam::enterData()
{
    int i;
    input();
    cout << "\nEnter marks in six subjects (out of 100):\n";
    for (i = 0; i < 6; i++)
    {
        cin >> marks[i];
    }
}

float exam::add()
{
    int i = 0;
    float sum = 0;
    for (i = 0; i < 6; i++)
    {
        sum = sum + marks[i];
    }
    return sum;
}

class result : public exam
{
    float totalMarks;
    float percentage;

public:
    void display()
    {
        totalMarks = add();
        percentage = totalMarks / 6;
        output();
        cout << "\nTotal Marks: " << totalMarks << "/600";
        cout << "\nPercentage : " << percentage << "%";
    }
};

main()
{
    result r;

```

```
    r.enterData();  
    r.display();  
    return 0;  
}
```

Output:

```
PS D:\sem 4\cpp\oops> cd "d:\sem 4\cpp\oops\" ; if ($?) { g++ inheritance16.cpp -o inheritance16 } ; if ($?) { .\inheritance16 }

Enter name: reeha

Enter roll no: 141

Enter marks in six subjects (out of 100):
95
99
97
98
100
96

NAME      : reeha
ROLL NO.  : 141
Total Marks: 585/600
Percentage : 97.5%
PS D:\sem 4\cpp\oops>
```

```
PS D:\sem 4\cpp\oops> cd "d:\sem 4\cpp\oops\" ; if ($?) { g++ inheritance16.cpp -o inheritance16 } ; if ($?) { .\inheritance16 }

Enter name: saba

Enter roll no: 2

Enter marks in six subjects (out of 100):
99
99
100
85
100
98

NAME      : saba
ROLL NO.  : 2
Total Marks: 581/600
Percentage : 96.8333%
PS D:\sem 4\cpp\oops>
```

Viva Questions

Q1) What is inheritance?

Ans.

Inheritance is one of the feature of Object Oriented Programming System(OOPs), it allows the child class to acquire the properties (the data members) and functionality (the member functions) of parent class.

Q2) What are advantages of inheritance?

Ans.

Code reusability: Now you can reuse the members of your parent class. So, there is no need to define the member again. So less code is required in the class.

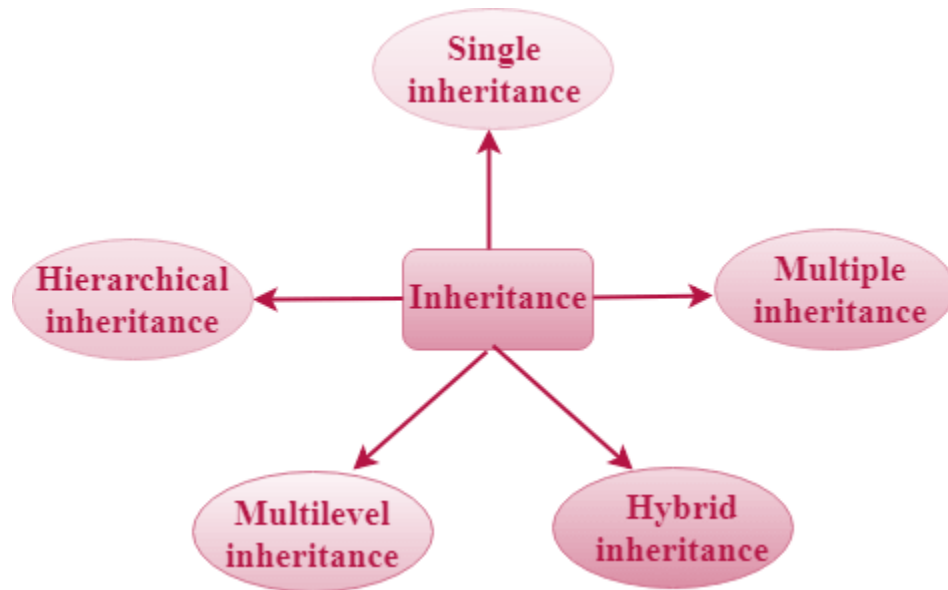
Q3) What are various types of inheritance?

Ans.

Various types of inheritance in C++ are:

- Single inheritance
- Multiple inheritance
- Hierarchical inheritance
- Multilevel inheritance

- Hybrid inheritance



Q4) How to make private member inheritable?

Ans.

The private member is not inheritable. If we modify the visibility mode by making it public, but this takes away the advantage of data hiding.

C++ introduces a third visibility modifier, i.e., protected. The member which is declared as protected will be accessible to all the member functions within the class as well as the class immediately derived from it.